



PATENT ABSTRACTS OF JAPAN

(11) Publication number: **07073107 A**(43) Date of publication of application: **17.03.95**

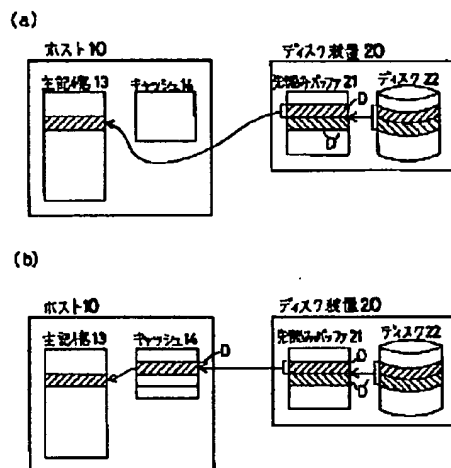
(51) Int. Cl.

G06F 12/08(21) Application number: **05220092**(71) Applicant: **PFU LTD**(22) Date of filing: **03.09.93**(72) Inventor: **IKEDA EIJI****(54) CONTROL METHOD FOR DISK SYSTEM****(57) Abstract:**

PURPOSE: To set data transfer to be efficient and to improve a cache hit rate by directly storing request data in a main storage at the time of sequential access and storing request data in a disk cache and the main storage at time except for sequential access.

CONSTITUTION: At the time of sequential access, a host computer 10-side stores request data D transmitted from a disk device with look-ahead buffer 20 into the main storage 13 without storing it into the disk cache 14. At time except for sequential access, the host computer 10-side stores request data D transmitted from the disk device with look-ahead buffer 20 into the disk cache 14 and stores it into the main storage 13. Thus, data transfer can be set efficient and the cache hit rate improves.

COPYRIGHT: (C)1995,JPO



(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開平7-73107

(43) 公開日 平成7年(1995)3月17日

(51) Int.Cl.⁶
G06F 12/08

識別記号
320

庁内整理番号
7608-5B

FI

技術表示箇所

審査請求 未請求 請求項の数 1 0L (全14頁)

(21) 出願番号 特願平5-220092
(22) 出願日 平成5年(1993)9月3日

(71) 出願人 000136136
株式会社ピーエフユー
石川県河北郡宇ノ気町字宇野気ヌ98番
地の2
(72) 発明者 池田 英二
石川県河北郡宇ノ気町字宇野気ヌ98番
地の2 株式会社ピーエフユー内
(74) 代理人 弁理士 京谷 四郎 (外1名)

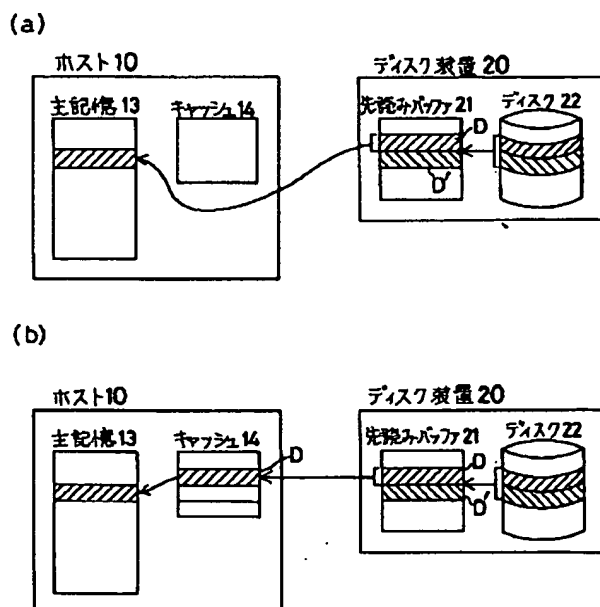
(54) 【発明の名称】 ディスク・システムの制御方法

(57) 【要約】

【目的】 本体側に設けられたディスク・キャッシュと、先読みバッファを有するディスク装置とを備えるディスク・システムにおいて、データ転送を効率よく行い得ると共に、キャッシュ・ヒット率を向上できるようにすること。

【構成】 シーケンシャル・アクセス時には、ディスク装置から送られて来た要求データをディスク・キャッシュを経由することなく、主記憶に転送する。シーケンシャル・アクセス以外の時は、ディスク装置から送られて来た要求データをディスク・キャッシュに格納すると共に、主記憶に転送する。

本発明の原理説明図



【特許請求の範囲】

【請求項1】 主記憶(13)と、
ディスク・キャッシュ(14)と、
先読みバッファ(21)およびディスク装置(22)を有する先
読みバッファ付きのディスク装置(20)とを具備するディ
スク・システムの制御方法であって、
シーケンシャル・アクセス時には、先読みバッファ付き
ディスク装置(20)から送られて来た要求データを、ディ
スク・キャッシュ(14)に格納することなく、主記憶(13)
に格納し、
シーケンシャル・アクセス以外の時には、先読みバッ
ファ付きディスク装置(20)から送られて来た要求データを
ディスク・キャッシュ(14)に格納すると共に、主記憶(1
3)に格納することを特徴とするディスク・システムの制
御方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、ホスト計算機側にディ
スク・キャッシュを備えると共に、ディスク装置の中に
先読みバッファを有するディスク・システムの制御方法
に関するものである。

【0002】

【従来の技術】図3は計算機システムの全体構成を示す
図である。同図において、10はホスト計算機、11は
中央処理装置、12はDMAコントローラ、13は主記
憶、14はディスク・キャッシュ、20は先読みバッ
ファ付きディスク装置、21は先読みバッファ、22はディ
スク装置をそれぞれ示している。

【0003】ホスト計算機10は、中央処理装置11や
DMAコントローラ12、主記憶13、ディスク・キャ
ッシュ14などを有している。ホスト計算機10には、
その他、ディスプレイ装置や、キーボード、プリンタ等
が接続される。先読みバッファ付きディスク装置20
は、先読みバッファ21やディスク装置22などを有し
ている。

【0004】ディスク・キャッシュ14は、先読みバッ
ファ付きディスク装置20と主記憶13の間に存在し、
先読みバッファ付きディスク装置20からのデータをそ
の中に保持できる構成となっている。もちろん、ディ
スク・キャッシュ14を使用せずに先読みバッファ付き
ディスク装置20からのデータを主記憶13へ直接転送す
ることも出来る。これらの制御はディスク制御プログラ
ムのDMAコントローラ12への指示によって行われ
る。

【0005】DMAコントローラ12は複数のチャンネ
ル(例えば、8チャンネル)を有しており、その内の一
つが主記憶13をアクセスするために使用され、その内
の一つがディスク・キャッシュ14をアクセスするため
に使用される。DMAコントローラ12と先読みバッ
ファ付きディスク装置20との間のデータ転送は、例えば

SCSIインタフェースを介して行われる。

【0006】図4はディスク・キャッシュの構成を示す
図である。ディスク・キャッシュは、キャッシュ・メモ
リ、キャッシュ管理テーブル、アクセス履歴テーブルな
どから構成されている。キャッシュ・メモリは、1MB
ないし16MB程度の大きさを有しており、ディスク内
データの一部コピーを保存する。1ブロックはディスク
装置と同サイズ(256バイトまたは512バイト)であ
る。

【0007】キャッシュ管理テーブルは、1項目が数バ
イト(例えば2バイト)であり、キャッシュ・メモリと
1対1に対応しており、現在、キャッシュ・メモリの中
にディスクの何れの部分のデータが保持されているかを
示すテーブルである。アクセス履歴テーブルは、過去の
アクセス履歴を保持するテーブルである。詳細は後述す
る。

【0008】図5はディスク・キャッシュ内ブロック格
納位置を説明する図である。キャッシュ・メモリのデー
タの格納位置およびそれと1対1に対応しているキャ
ッシュ管理テーブルの位置を決定する方式としては、ハッ
シュ方式が使用される。具体的に言うと、キャッシュ・
メモリは1MBないし16MB程度とディスクに比較し
てかなり小さいものなので、ディスク内データの或る一
部分しか格納できないが、或るディスクのブロック位置
(アドレス)が判明すると、一意にキャッシュの位置が
決定するという方式である。

【0009】動作としては、例えばキャッシュ・メモリ
が1MBであるとし、ディスクの1ブロックが512バ
イトであるとする、1MB(1048576バイト)
 $\div 512 \text{ バイト} = 2048$ ブロック分のデータがキャ
ッシュに格納できることになるが、この数のことをエント
リ数と言う。図5のようにキャッシュ内を、使用してい
くとする。同図において、数字はディスクのブロック・
アドレスを示す。つまり、ブロック0、ブロック204
8、ブロック4096、…はAの位置に格納される。同
様に、ブロック1、ブロック2049、…はBの位置に
格納され、ブロック2、ブロック2050、…はCの位
置に格納されるように、使用していくとする。これは、
ディスクのブロックXが与えられた場合、Xをエントリ
数(この場合2048)で割った余りによってaから何
段目かが決定するという方式である。

【0010】すなわち、 $X \div (\text{エントリ数}) = Y \text{ 余り } n$
とすると、このnがaから何段目かを示す数になる。デ
ィスクのブロック・アドレスが2050の場合は、 $2050 \div 2048 = 1 \text{ 余り } 2$ となり、アドレス2050の
ブロックは、キャッシュ・メモリのaから数えて2段目
のCの位置に格納される。アドレスが4096の場合
は、 $4096 \div 2048 = 2 \text{ あまり } 0$ となり、アドレス
4096のブロックはキャッシュ・メモリのaから数え
て0段目のAの位置に格納される。このような方式をハ

ッシュ方式と言い、簡単なアルゴリズムでキャッシュの格納位置を決定することが出来る。

【0011】図6は先読みバッファ付きディスク装置の構成例を示す図である。同図において、20は先読みバッファ付きディスク装置、21は先読みバッファ、22はディスク装置、23はコントローラをそれぞれ示している。先読みバッファ21は、256KBないし512KB程度の大きさを有している。コントローラ23は、例えばマイクロプロセッサから構成されている。

【0012】図7はディスク制御プログラムの制御フローを示す図である。先ず、ディスクからデータをリードする場合について説明する。ソフトウェアから、ディスク・データのリード要求があった場合、ディスクの読取り開始ブロック・アドレスとブロック数が渡される。これに基づき、ディスク制御プログラムはキャッシュ管理テーブルをサーチして、キャッシュ・メモリ内に該当データが存在するかどうかをチェックする。存在すればヒットと言い、存在しなければミスヒットという。

【0013】ヒットの場合、キャッシュ・メモリからデータを主記憶へ転送し、ディスク装置にはアクセスしない。これによってディスク装置を高速に見せている。ミスヒットの場合、アクセス履歴によって、シーケンシャル・アクセス（順編成ファイルへのアクセス等が此れに該当する）かどうかを判断し、シーケンシャルアクセスの場合は先読み分を含めた分のリードを先読みバッファ付きディスク装置に要求し、そうでなければ要求分のみのリードを先読みバッファ付きディスク装置に要求する。

【0014】先読みバッファ付きディスク装置から転送されたデータは、ディスク・キャッシュの該当位置に格納されると同時に要求分は主記憶へ転送される。その後で、キャッシュ管理テーブルの更新（新たに先読みバッファ付きディスク装置からキャッシュに転送されたデータに対して）が行われる。これで、一回のソフトウェアからの起動が終了する。この後のソフトからの起動で、先程と同じデータの要求があった場合、もしくはシーケンシャル・アクセスで後続のデータの要求があった場合は、キャッシュにヒットする可能性が高い。

【0015】次に、ディスクにデータをライトする場合の動作について説明する。ライト時にはリードのようにヒット／ミスヒットはない。ソフトウェアから主記憶データのディスクへのライト要求（格納指示）があった場合、リードの場合と同様にディスクの書き込み開始ブロック・アドレスとブロック数が渡される。これに基づき、ディスク制御プログラムは、ディスク・キャッシュ上の格納位置を決定し、主記憶上のデータをディスク・キャッシュに転送すると同時に先読みバッファ付きディスク装置にもデータを書き込むための処理を行う。両方の転送が終了した時点でキャッシュ管理テーブルの該当項目を更新する。これで、ソフトウェアからの一回のライト

起動が終了する。この後のソフトウェアからの起動で、先ほどライトしたデータと同じデータのリード要求があった場合は、ディスク・キャッシュにヒットする可能性が高い。

【0016】図8は従来のホスト側ディスク・キャッシュの先読み動作を説明するための図である。同図において、①はディスク装置からキャッシュへの先読み分も含めたリード、②はディスク・キャッシュから主記憶への要求分のみの転送、③はディスク・キャッシュにおける先読み分、④は先読みバッファにおける先読み分をそれぞれ示している。

【0017】いま、ソフトウェアからディスク・データのリード要求があり、ディスクの読取り開始ブロック・アドレス=X、ブロック数=Y1、主記憶アドレス=Zであったと仮定する。ホスト計算機10（実際にはディスク制御プログラム）は、このリード要求で指定されたデータがディスク・キャッシュ14にあるか否かを調べ、ない場合にはアクセス履歴を参照して、シーケンシャル・アクセスか否かを調べる。シーケンシャル・アクセスの場合には、先読み分をY2とし、開始ブロック・アドレス=X、ブロック数=Y1+Y2をパラメータとして持つリード・コマンドを先読みバッファ付きディスク装置20に送る。

【0018】先読みバッファ付きディスク装置20は、リード・コマンドを受け取ると、リード・コマンドで指定されたブロックが先読みバッファ21にあるか否かを調べる。ある場合には、ブロック・アドレス=Xを先頭とする(Y1+Y2)ブロックを先読みバッファ21から読み出してホスト計算機10に送る。ホスト計算機10（実際にはDMAコントローラ12）は、ブロック・アドレス=Xを先頭とする(Y1+Y2)ブロックをディスク・キャッシュ14に書き込み、ブロック・アドレス=Xを先頭とするY1ブロックを主記憶13のアドレスZ以降の領域に書き込む。

【0019】リード・コマンドで指定されたデータが先読みバッファ21にない場合は、先読みバッファ付きディスク装置20は、リード・コマンドで指定されたデータを含む所定数（例えば1000）のブロックをディスク装置22から読み出して先読みバッファ21に格納すると共に、ブロック・アドレス=Xを先頭とする(Y1+Y2)ブロックを読み出してホスト計算機10に送る。それ以後の動作は前述した通りである。

【0020】図9はディスク装置からデータをリードする場合のホスト計算機の処理を示す図である。先読みバッファ付きディスク装置からデータをリードする場合、ソフトウェアからは、

① ディスクの読取り開始ブロック・アドレス

② ブロック数

③ 主記憶のメモリ・アドレス

の3つの情報が渡される。このうち、①と②の情報は、

リード・コマンドのパラメータとして、先読みバッファ付きディスク装置に渡される。先読みバッファ付きディスク装置から該当のデータがインタフェース・バスを介してホスト計算機に流れて来る。このとき、③の情報を

用いてDMAコントローラによって主記憶に展開する。
【0021】図10は先読みバッファ付きディスク装置の先読みバッファの構造を示す図である。先読みバッファの全体のサイズは256KBないし512KB程度であり、これをセグメント数と呼ばれる分割数で分割して保持している。セグメント数は通常は4ないし8程度であり、したがって、1セグメント当たり32ないし128KB程度と言うことになる。

【0022】この先読みバッファは、2ポート構成となっており、ディスク媒体から読み取って先読みバッファに格納するポートと、インタフェース・バスから外部へ送出するポートとを持っており、媒体からデータを読み取りながら外部へ送出することが出来るようになっている。なお、どれかを上書きして使用することになるが、どれを使用するかはLRU (Least Recently Used) という有名なアルゴリズムを使用して決定する。つまり、最も古い(最も以前の)アクセスで使用されたバッファを潰して使用し、今度はそのバッファを最新のものとして登録しておくので、常になるべく新しいデータをバッファに保持することが出来る仕組みである。

【0023】1つの先読みバッファのサイズは32KBないし128KB程度であるが、32KBの場合を想定する。ディスクの1ブロックのサイズを512バイトとすると、32KB (32768バイト) ÷ 512バイト = 64ブロック分のデータを1個の先読みバッファの中に保持できることになる。図示しないが、各先読みバッファに対応して管理テーブルが存在し、各管理テーブルは、対応する先読みバッファの各ブロック格納域にどのブロック・アドレスのデータが格納されているかを示す情報を保持している。

【0024】図11および図12はリード時の先読みバッファ付きディスク装置の処理を示す図である。ホストからリード・コマンドが送られてくると、ブロック・アドレスとブロック数が渡されるので、先読みバッファ付きディスク装置内のコントローラは、ディスク媒体の該当位置へヘッドを移動する(シークという)。そして、目的のブロックを見つけて、データを先読みバッファの一つへ格納していく。これと同時に、外部送出用のポートから外部へデータを送出していく。

【0025】このようにホスト計算機からのリード・コマンドに基づいて該当データを転送していくわけであるが、例えば、ホスト計算機の要求ブロックが10ブロックだったと仮定する。1個の先読みバッファ(1セグメント)は、64ブロック分のデータを保持できるわけであるから、ホスト計算機へのデータ転送が終了した後も54ブロック分のデータを継続して先読みして、先読み

バッファに格納していく。その後のホスト・アクセスが先程の後続データであれば、先読みバッファにヒットすると言う仕組みである。

【0026】先読み中にホストからの新たなリード・コマンドを受け取った場合は、図12のフロー中の最後の☆印の箇所チェックが入り、そのコマンドの要求データが先程の後続データであればそのまま先読みバッファの中に入りつつあるデータをホスト計算機へ転送し、後続データでなければ先読み動作は速やかに中止して、新たなリード・コマンドの処理を行う。これによって、性能劣化を防いでいる。

【0027】図13および図14はシーケンシャル・アクセス判断処理を示す図である。図7の制御フローにおいて、アクセス履歴を参照して、シーケンシャル・アクセスか否かの判断を行っているが、過去16回までのアクセス履歴を記憶しておくために、“アクセス履歴テーブル”と呼ばれるテーブルを持っている。これは、ソフトウェアからのリード指示が発行された場合、その開始ブロック・アドレスとブロック数を加算したもの(4バイト)を16通り覚えておくもので、64バイトのテーブルとなっている。図13および図14はシーケンシャル・アクセス判断処理のフローを示す。図示のアルゴリズムを使用することによって、今回のアクセスがシーケンシャルアクセスかどうか、SEQフラグのON/OFFによって判明する。さらに今回のアクセス状況が新たにテーブルに追加されることになる。

【0028】図15はアクセス履歴テーブルの遷移の例を示す図である。図15(a)は初期状態を示す。初期状態では、アクセス履歴テーブルの各項目(行)には、FFFFFFFFFが格納されている。ディスクの最大容量の関係上、FFFFFFFFFはあり得ないため、この値を設定している。00000000にすると、00000000が開始ブロックとなるアクセスが、常にシーケンシャル・アクセスと誤認識されてしまう。

【0029】アクセス履歴テーブルが図15(a)の状態の下でブロック00000000から10ブロックのリード要求があった場合には、アクセス履歴テーブルの内容は図15(b)のように変化する。シーケンシャル・アクセス判断のアルゴリズムを使用すると、テーブルの項目内に00000000が存在しないため、シーケンシャル・アクセスでないと判断される。最初のアクセスは常にシーケンシャル・アクセスでないと判断されるが、最初だけなので、頻度は最小限である。

【0030】アクセス履歴テーブルが図15(b)の状態の下でブロック00000010から10ブロックのリード要求があった場合には、アクセス履歴テーブルの内容は図15(c)のように変化する。アクセス履歴テーブルの項目内に00000010と言う項目が存在するため、シーケンシャル・アクセスと判断される。図15(b)のテーブル状態から00000010の箇所が更新

されて、 $00000010+10=00000020$ となり、図15(c)のようになる。

【0031】その後、様々なアクセスがあって、アクセス履歴テーブルの内容が図15(d)のようになったと仮定する。アクセス履歴テーブルが図15(d)の状態の下でブロック00222222から8ブロックのリード要求があった場合には、アクセス履歴テーブルの内容は図15(e)のように変化する。アクセス履歴テーブルの項目内に00222222と言う項目が存在するため、シーケンシャル・アクセスと判断される。図15(d)のテーブルの状態から00222222の箇所が更新されて、 $00222222+8=0022222A$ となり、且つその項目が先頭(最新箇所)となり、古い項目は後ろにずれる。

【0032】アクセス履歴テーブルが図15(e)の状態の下でブロック0000ABCDから2ブロックのリード要求があった場合には、アクセス履歴テーブルの内容は図15(f)のように変化する。アクセス履歴テーブルの項目内に0000ABCDと言う項目が存在しないため、シーケンシャル・アクセスでないと判断される。図15(e)のテーブル項目に $0000ABCD+2=0000ABCF$ の項目が追加され、且つ其の項目が先頭(最新箇所)となり、古い項目が後ろにずれる。このとき、最も古い00000020が追い出される(アクセス履歴テーブルから削除される)。

【0033】従来の技術では、ディスク・キャッシュの中に、

- i) アクセス要求のあったブロックのデータを保持する。(ソフトの“データの再使用”と言う性質を利用)
 - ii) アクセス要求のあったブロックの後続データを先読みして保持する。(ソフトの“参照の局所性”と言う性質を利用)
- などの動作を行い、ディスク装置を高速に見せている。しかし、最近のディスク装置は、装置内に先読みバッファを装備しており、ii)の機能は意味がなくなっている。

【0034】

【発明が解決しようとする課題】上述した従来の技術には、以下のような問題点がある。

(a) 以降のリード要求で、ディスク・キャッシュ内の先読みのデータがヒットされることが予測されるが、このときディスク・キャッシュのデータを主記憶へ転送する時間と、先読みバッファ付きディスク装置の先読みバッファのデータをホスト側キャッシュを介さずに主記憶へ転送する時間は同等であり、ディスク・キャッシュの効果はなくなる。

(b) そればかりか、先読みバッファ付きディスク装置からの先読みデータの転送でディスク・キャッシュ内の旧データが消失してしまうため、ディスク・キャッシュの使用効率が悪く、結果的に全体を通したヒット率が低下

する。すなわち、シーケンシャル・アクセス時は、ホスト側キャッシュがあってもなくても性能は変わらない。また、ディスク装置側からの転送速度が高速になると、ディスク・キャッシュが付加されている方が遅くなると言う性能逆転現象が発生する。

【0035】次にシーケンシャル・アクセスでない場合を考える。

(1) ディスク装置側のバッファが主に先読み用に考えられていること。

(2) (ホスト側キャッシュ容量) > (ディスク側先読みバッファ容量) であること。

(3) ディスク・アクセス時は、どのような場合でも、先読みバッファ付きディスク装置の先読みバッファを使用してしまうこと。

等の理由により先読みバッファのデータ保持性は低いと言える。すなわち、シーケンシャル・アクセス以外の場合は、ディスク装置側の先読みバッファにキャッシュとしての効果は期待できない。なお、シーケンシャル・アクセスの場合は、一度使用されたデータは二度と使用しない場合が多いので、データ保持性は低くても問題は無い。

【0036】本発明は、この点に鑑みて創作されたものであって、データ転送を効率良く行い得るとともに、キャッシュ・ヒット率を向上できるようになったディスク・システムの制御方法を提供することを目的としている。

【0037】

【課題を解決するための手段】図1は本発明の原理説明図である。請求項1のディスク・システムの制御方法は、主記憶(13)と、ディスク・キャッシュ(14)と、先読みバッファ(21)およびディスク装置(22)を有する先読みバッファ付きのディスク装置(20)とを具備するディスク・システムの制御方法であって、シーケンシャル・アクセス時には、先読みバッファ付きディスク装置(20)から送られて来た要求データを、ディスク・キャッシュ(14)に格納することなく、主記憶(13)に格納し、シーケンシャル・アクセス以外の時には、先読みバッファ付きディスク装置(20)から送られて来た要求データをディスク・キャッシュ(14)に格納すると共に、主記憶(13)に格納することを特徴とするものである。

【0038】

【作用】図1(a)を参照して、シーケンシャル・アクセス時の動作について説明する。先読みバッファ付きディスク装置20は、ホスト計算機からのリード要求を受け取ると、ディスク装置22から要求データDと、それに続く先読分のデータD'を読み出し、要求データDと先読み分のデータD'を先読みバッファ21に格納し、要求データDをホスト計算機側に転送する。ホスト計算機側は、シーケンシャル・アクセス時には要求データDをディスク・キャッシュ14に格納することなく、受け取

った要求データDを主記憶13に格納する。

【0039】図1(b)を参照して、シーケンシャル・アクセス以外の時の動作について説明する。先読みバッファ付きディスク装置20は、ホスト計算機からのリード要求を受け取ると、ディスク22から要求データDと、それに続く先読みのデータD'を読み出し、要求データDと先読み分のデータD'を先読みバッファ21に格納し、要求データDをホスト計算機側に転送する。ホスト計算機側は、シーケンシャル・アクセス以外の時には要求データDをディスク・キャッシュ143に格納すると共に、要求データDを主記憶13に格納する。

【0040】本発明では、シーケンシャル・アクセス時にはディスク・キャッシュにデータを格納せず、シーケンシャル・アクセス以外の時には要求分のみをディスク・キャッシュに格納しているので、ディスク・キャッシュ内の旧データの消失を最小限にすることができ、ディスク・キャッシュのデータはその後のシーケンシャル・アクセス時でも消失しない。また、数回のディスク・アクセス後、同一のデータの再要求があった場合には、このディスク・キャッシュにより、ヒットが期待できる。もし、このようなディスク・キャッシュが無く、ディスク内の先読みバッファのみでキャッシュを構成した場合は、数回のディスク・アクセス後、同一データの再要求があった場合は、数回のディスク・アクセス中にデータが消えてしまうので、ヒットが期待できない。

【0041】

【実施例】図2は本発明における動作フローを示す図である。本発明を実施するためのハードウェア構成は、従来例と同じである。本発明と従来例の相違は、ディスク制御プログラムの一部が相違するだけである。すなわち、図7において、「該当アクセス履歴更新」→「先読み分を含めてディスク起動。データはキャッシュに格納」→「キャッシュ管理テーブル更新」→「要求分のみキャッシュから主記憶転送」と言う処理の流れがあるが、本発明はこの処理の流れの部分を変更したものである。ステップS1では、ソフトウェアからのリード要求を受け取る。ステップS2では、過去何回かのアクセス履歴を参照する。ステップS3では、シーケンシャル・アクセスか否かを調べる。YESの場合はステップS4に進み、NOの場合はステップS7に進む。

【0042】ステップS4では、該当アクセス履歴を更新する。ステップS5では、ディスクにリード起動を送る。この場合は、ホスト側のディスク・キャッシュは使用しない。ステップS6では、ディスクからのデータをそのまま主記憶へ転送する。

【0043】ステップS7では、アクセス履歴を追加す*

*る。ステップS8では、ディスクにリード起動を送る。ステップS9では、ソフトからの要求分のみキャッシュにデータ格納する。ステップS10では、キャッシュから主記憶へデータ転送する。

【0044】

【発明の効果】以上の説明から明らかなように、本発明によれば、キャッシュ・ヒット率を向上できると共に、効率よくデータを転送することが可能となる。

【図面の簡単な説明】

【図1】本発明の原理説明図である。

【図2】本発明における動作フローを示す図である。

【図3】計算機システムの全体を示す図である。

【図4】ディスク・キャッシュの構成を示す図である。

【図5】ディスク・キャッシュ内ブロック格納位置を説明するための図である。

【図6】先読みバッファ付きディスク装置の構成を示す図である。

【図7】ディスク制御プログラムの制御フローを示す図である。

【図8】従来のホスト側ディスク・キャッシュの先読み動作を説明する図である。

【図9】ディスク装置からデータをリードする場合のホスト計算機の処理を示す図である。

【図10】ディスク装置内の先読みバッファの構造を示す図である。

【図11】リード時のディスク装置の処理を示す図である。

【図12】リード時のディスク装置の処理（続き）を示す図である。

【図13】シーケンシャル・アクセス判断処理を示す図である。

【図14】シーケンシャル・アクセス判断処理（続き）を示す図である。

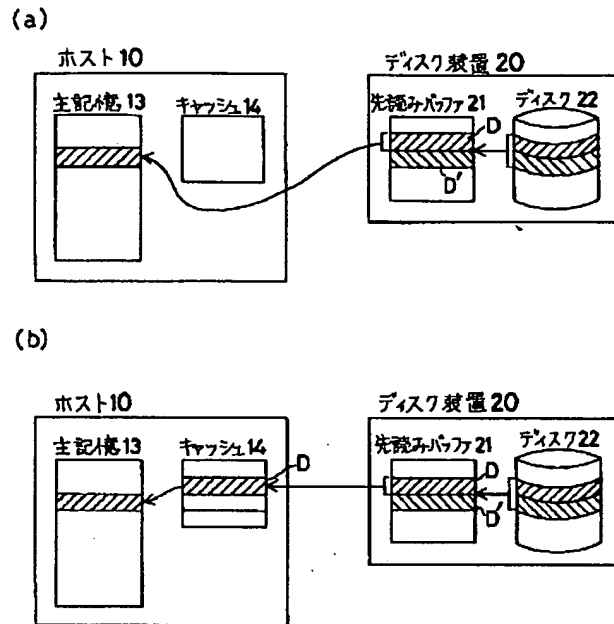
【図15】アクセス履歴テーブルの遷移の例を示す図である。

【符号の説明】

- 10 ホスト計算機
- 11 中央処理装置
- 12 DMAコントローラ
- 13 主記憶
- 14 ディスク・キャッシュ
- 20 先読みバッファ付きディスク装置
- 21 先読みバッファ
- 22 ディスク装置
- 23 コントローラ

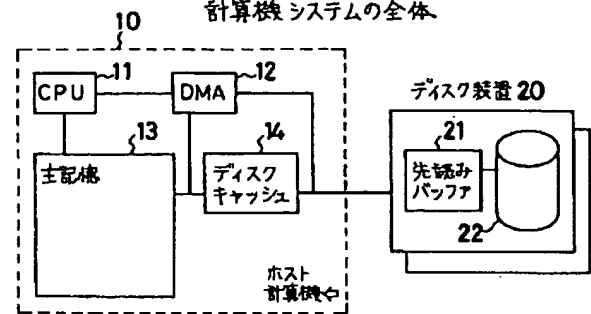
【図1】

本発明の原理説明図



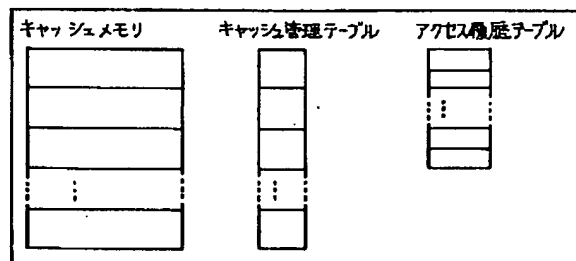
【図3】

計算機システムの全体



【図4】

ディスク・キャッシュの構成



【図5】

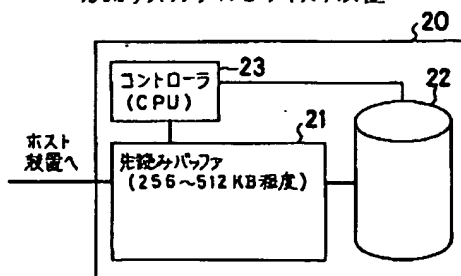
ディスク・キャッシュ内ブロック格納位置

a ⇒

0	2048	4096	⋮	⇐A
1	2049	⋮	⋮	⇐B
2	2050	⋮	⋮	⇐C
3	2051	⋮	⋮	⇐D
⋮	⋮	⋮	⋮	
2046	4094	⋮	⋮	
2047	4095	⋮	⋮	

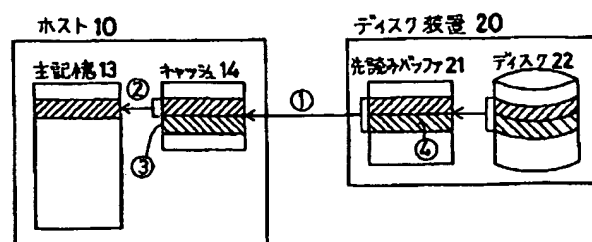
【図6】

先読みバッファ付きディスク装置



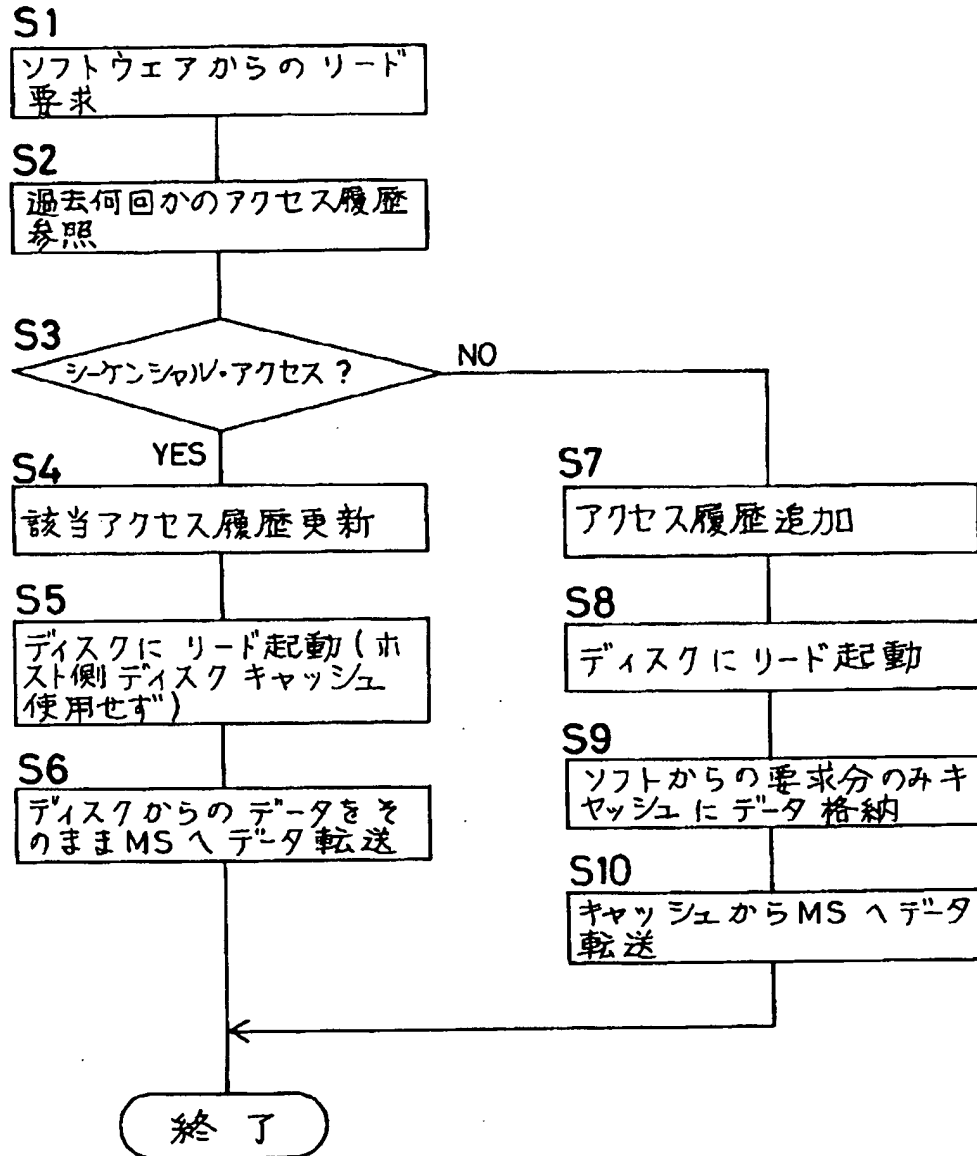
【図8】

従来のホスト側ディスクキャッシュの先読み動作



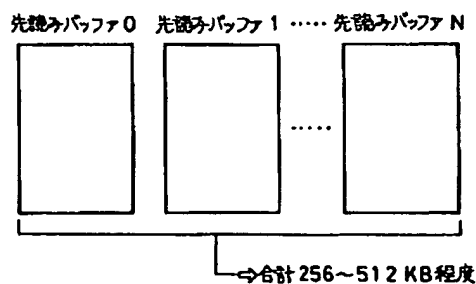
【図2】

本発明における動作フロー



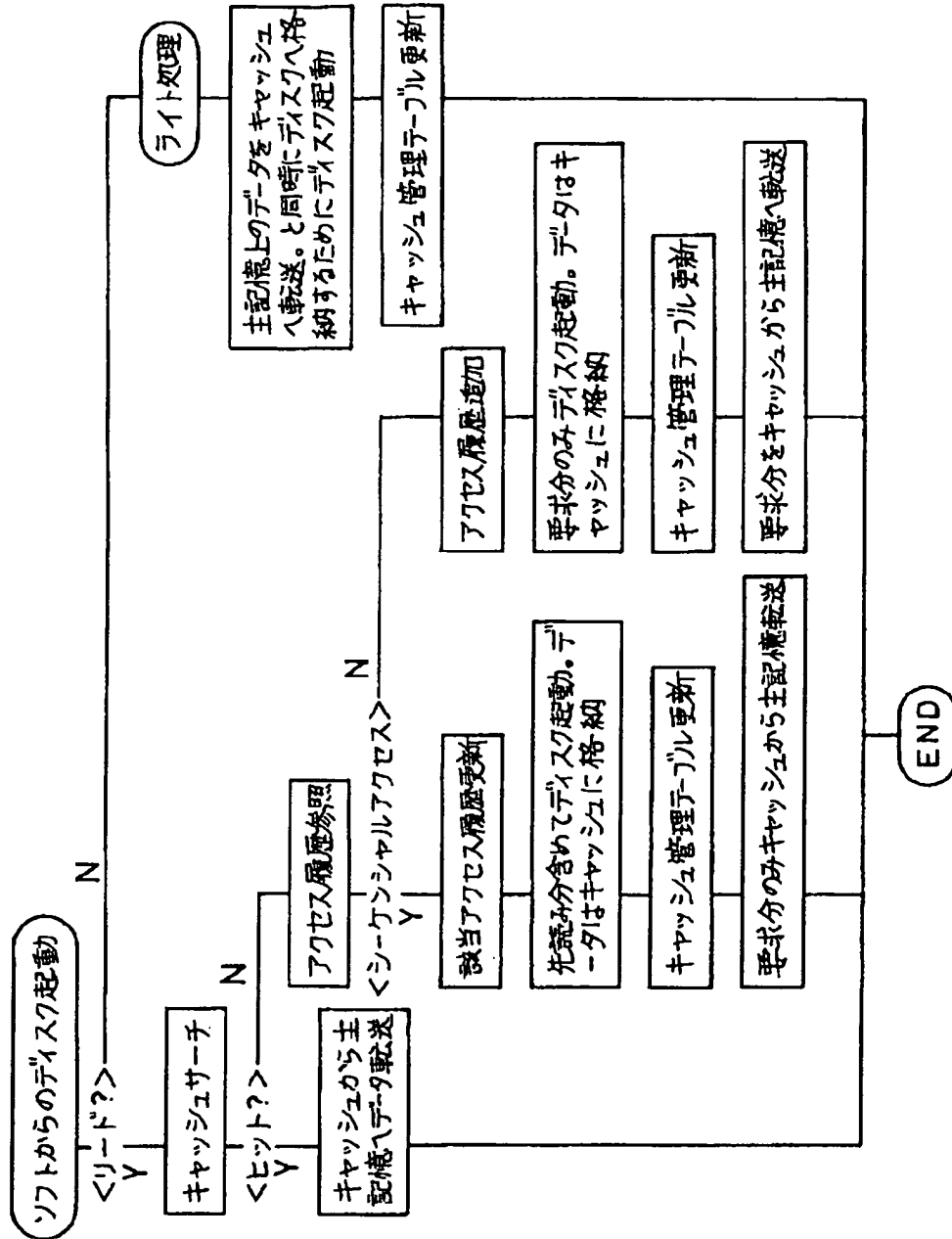
【図10】

ディスク装置内の先読みバッファの構成



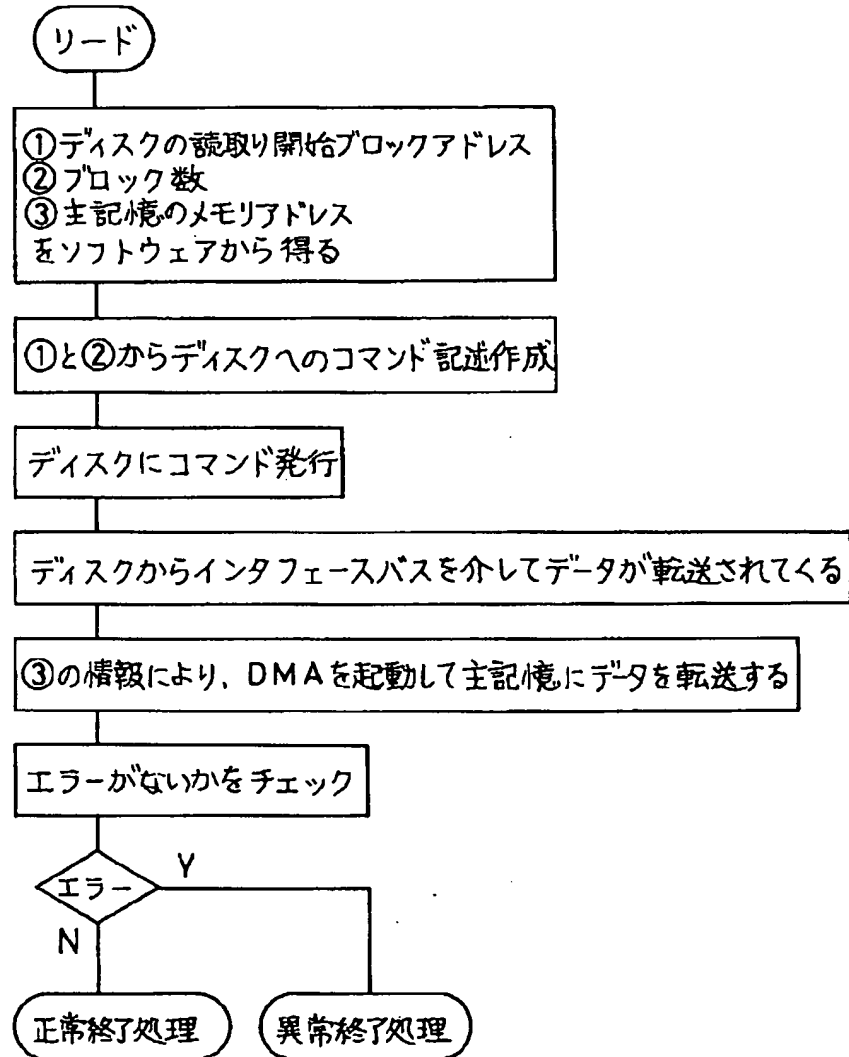
【図7】

ディスク制御プログラムの制御フロー



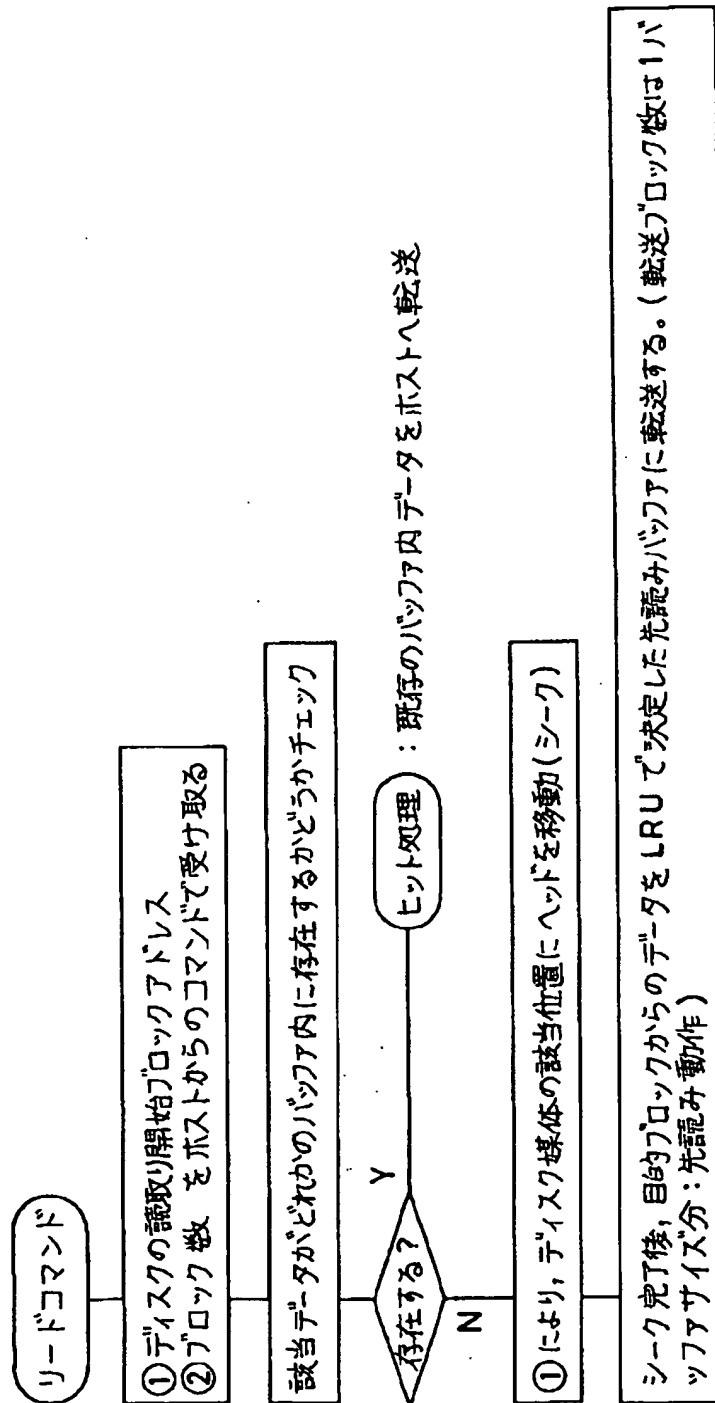
【図9】

ディスク装置からデータをリードする場合のホスト計算機の処理



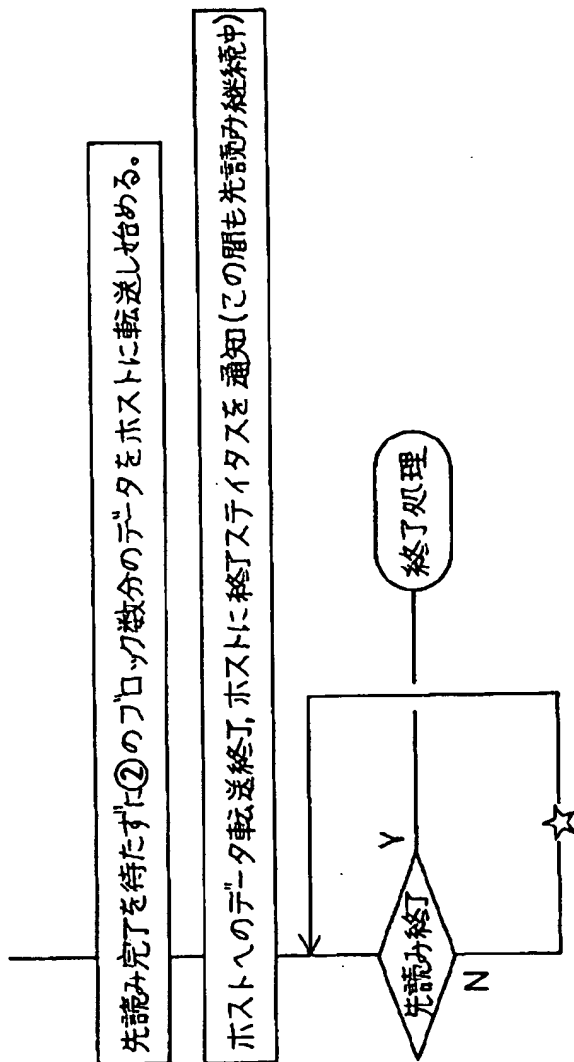
【図11】

リード時のディスク装置の処理



【図12】

リード時のディスク装置の処理(続き)



【図15】

アクセス履歴テーブルの遷移の例

(a) 初期状態

FFFFFFFF
FFFFFFFF
FFFFFFFF
⋮
FFFFFFFF

(d) その後、様々なアクセスがあり、(d)のような状態になったとする。

00087654
01234567
00222222
⋮
00000020

(b) ブロック00000000から10
ブロックのリード要求があった場合

00000010
FFFFFFFF
FFFFFFFF
⋮
FFFFFFFF

(e) ブロック00222222から8
ブロックのリード要求があった場合

0022222A
00087654
01234567
⋮
00000020

(c) ブロック00000010から10
ブロックのリード要求があった場合

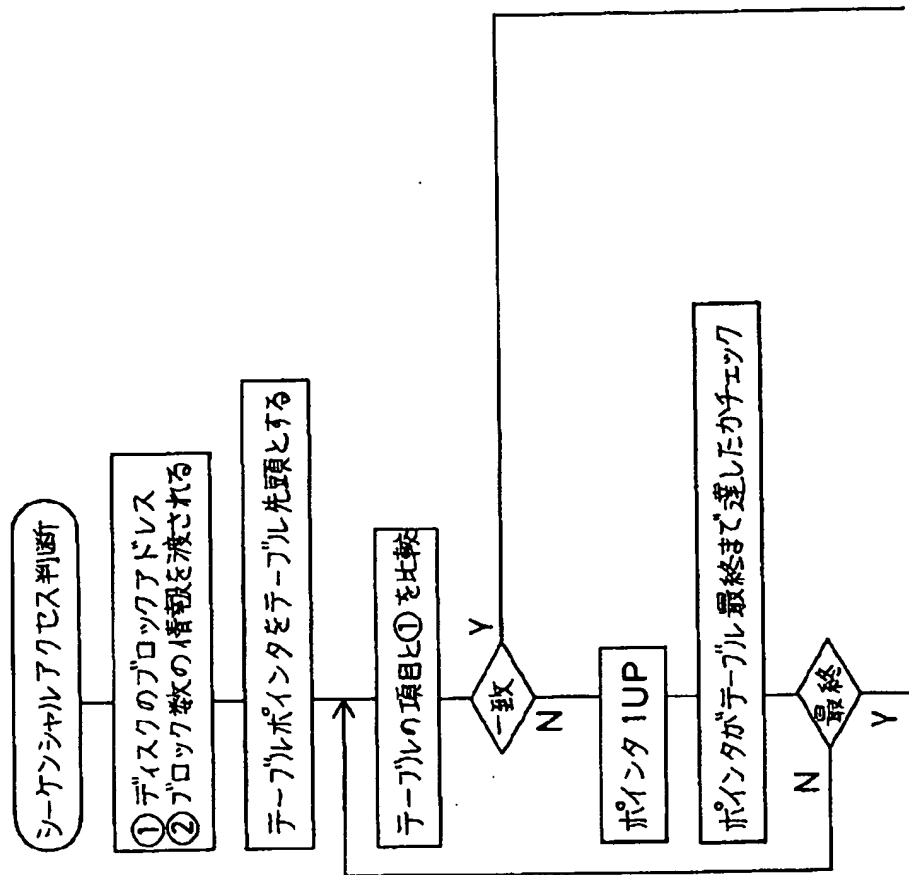
00000020
FFFFFFFF
FFFFFFFF
⋮
FFFFFFFF

(f) ブロック0000ABCDから2
ブロックのリード要求があった場合

0000ABCF
0022222A
00087654
⋮
00000540

【図13】

シーケンシャル・アクセス判断処理



【図14】

シーケンシャル・アクセス判断処理(続き)

